

Dimension: Syntax

```
add dimension NAME
description VALUE
  icon FILENAME WITH
  PATH
  [!inot]hidden
property NAME [to
ADMINTYPE NAME] [value
STRING]
unit UNITNAME label
UNITLABEL multiplier
MULTIPLIER offset OFFSET
[!inot]default
property NAME [to
ADMINTYPE NAME] [value
STRING]
setting NAME VALUE
history STRING;
```

Type: Syntax

```
add type NAME
description VALUE
icon FILENAME WITH PATH
attribute NAME {, NAME}
derived TYPE NAME
abstract [true|false]
method PROG NAME {, PROG
NAME}
trigger EVENT TYPE
{action|check|override}
PROG NAME [input ARG
STRING]
[!|not]hidden
property NAME [to
ADMINTYPE NAME] [value
STRING]
history STRING;
```

Format: Syntax

```
add format NAME
description VALUE
icon FILENAME_ WITH PATH
creator NAME (Not Used)
```

Format: Syntax (cont)

```
type NAME (Not Used)
version VALUE
[!Inot]hidden
suffix VALUE
mime VALUE
property NAME [to
ADMINTYPE NAME] [value
STRING]
history STRING;
```

Group: Syntax

```
add group "group_name"
assign person "person
name"
assign role "role_name"
description "group_des-
cription"
icon [path] child "gro-
up_name"
parent "group_name"
[not]hidden;
```

Person: Syntax

```
add person "person_name"
access "user access
values"
admin "bus and/or system
access rights"
address "address_value"
assign "groupe role"
assign "role_name"
comment "comment_text"
enable/disable email
enable/disable iconmail
email "email_address"
fax "fax_number"
fullname "fullname_val-
ue"
```

Person: Syntax (cont)

```
phone "phone number"
site "default site"
vault "vault name"
password "value"
no password
disable password
type of user
[not]hidden;
```

Dimension: Example

```
add dimension Weight
description "Weight of
the item"
unit g
default
label Grams
unitdescription "Weight
in Grams (Default)"
multiplier 1.0
offset 0.0
unit Kg
label Kilograms
unitdescription "Weight
in Kilograms"
multiplier 1000.0
offset 0.0;
```

Type: Example

```
add type PART
abstract true
description "Part for a
computer"
attribute Cost,Source,-
Weight;
Or
add type Customer
```

Type: Example (cont)

```
description "Customer
information"
icon <path of the
directory> /HCCgifs/Cus-
tomer.gif
abstract false attribute
"Account Number"
attribute Address1
attribute Address2
attribute Address3
attribute City
attribute ID
attribute State
attribute "Zip Code";
```

Format: Example

```
add format Word
version "Office2010"
description "Microsoft
Word"
suffix ".docx";
```

Group: Example

```
add group Manufacturing
parent "HCC"
assign person "Joe
Jackson"
assign person "Dave
Matthews";
```

Person: Example

```
add person Joe
access all
email joe
fullname "Joe Jackson"
type full,business
vault Engineering
no password
```

Person: Example (cont)

```
icon <path of the
directory> /Icons/Jo-
e.gif
assign role President
assign group Manufa-
cturing
assign group Sales
assign group "HVV Corp";
```

Policy Example (1/2)

```
add policy Production
enforce
description "Policy to
govern
PARTS,,
sequence A,B,C,...
store Production
type PART
format Excel
format Word default-
tformat Word
state Planned
version false
revision false
promote false
checkouthistory false
owner all
public all
state Active
version false
revision true
promote false
checkouthistory false
owner all
public read,show
```

Attribute: Syntax

```
add attribute NAME
description VALUE
icon FILENAME_ WITH PATH
type [integer, string,
real, boolean, date]
default VALUE range =
RANGE_ITEM
[not]hidden
[not]multiline
[not]rangevalue
[not]multivalue
dimension DIMENSION
NAME;
trigger modify PROG_TYPE
PROG NAME [input ARG_ST-
RING];
```

Relationship: Syntax

```
add relationship
"relationship_name"
description "relations-
hip_description"
icon [path]_name
attribute ATTRIBUTE NAME
trigger EVENT_TYPE
TRIGGER TYPE PROG_NAME
[!] [not]preventdupl-
icates
derived RELATIONSHIP
NAME
abstract [true|false]
[!] [not]hidden
property NAME [to
ADMINTYPE NAME] [value
STRING]
history STRING
from
type "from types"
meaning "meaning_text"
```

Relationship: Syntax (cont)

```
cardinality [one, many]
revision [none, float,
replicate]
clone [none, float,
replicate]
[not] propagatemodify
to
type "to_types"
meaning "meaning_text"
cardinality [one, many]
revision [none, float,
replicate]
clone [none, float,
replicate]
[not] propagatemodify;
```

Role: Syntax

```
add role "role name"
assign person "person-
_name"
assign group "group_-
name"
description "descripti-
on_name"
icon [path] child "role
name"
parent "role name"
[not]hidden;
```

Policy: Syntax (1/2)

```
add policy NAME
description VALUE
icon FILENAME_WITH_PATH
type TYPE_NAME (,TYPE-
_NAME)
type all
```

Policy: Syntax (1/2) (cont)

```
format FORMAT_NAME
(,FORMAT_NAME)
format all
defaultformat FORMAT-
_NAME
sequence REVISION_S-
EQUENCE
[not]enforce
store STORE_NAME
allstate USER_ITEM
ACCESS_ITEM (,USER_ITEM
ACCESS_ITEM)
[!] [not]hidden
property NAME [to
ADMINTYPE NAME] [value
STRING]
history STRING
```

Policy Example (2/2)

```
state Obsolete
version false
revision true
promote false
checkouthistory false
owner all
public read,show;
modify policy "Produc-
tion"
state "Planned"
add signature "Ready"
approve "Vice Presid-
ent"
reject "Vice President"
state "Active"
add signature "Obsolete"
approve "Vice Presid-
ent"
reject "Vice Presid-
ent";
```

Attribute: Example

```
add attribute "Screen
Size in Inches"
description "The
diagonal measure of a
monitor screen in
inches"
type string default 15
range = 15
range = 17
range = 19
range = 21;
```

Relationship: Example

```
add relationship "System
Box Assembly"
description "Relati-
onship for assembling
the major components of
the System Box"
attribute "Quantity"
from
type "System Box"
meaning "Is comprised
of..."
cardinality many
revision replicate
clone none
to
type "SYSTEM BOX PART"
meaning "is a component
of..."
cardinality many
revision float
clone none;
```

Program: Syntax

```
add program NAME
description VALUE
icon [path]
java|external|mgl
execute immediate|de-
ferred
execute user USER NAME
[!|not] needsbusines-
subject
[!|not]downloadable
[!|not]pipe
[!|not]pooled
[!|not]hidden
rule NAME
property NAME [to
ADMIN|TYPE NAME] [value
STRING]
history STRING
code "command or code";
```

Role: Example

```
add role Shipper
description "Shipper in
the company";
```

Policy: Syntax (2/2)

```
state STATE_NAME
action COMMAND
check COMMAND
notify USER_NAME {,USER-
_NAME} message VALUE
notify signer message
VALUE
route USER_NAME message
VALUE
owner ACCESS_ITEM
{,ACCESS_ITEM}
```

Policy: Syntax (2/2) (cont)

```
public ACCESS_ITEM
{,ACCESS_ITEM}
user USER_NAME ACCESS-
_ITEM {,ACCESS_ITEM}
[filter EXPRESSION]
signature SIGN_NAME
[SIGNATURE_ITEM {,
SIGNATURE_ITEM}]
stateproperty NAME [to
ADMIN|TYPE NAME] [value
STRING]
trigger EVENT_TYPE
TRIGGER_TYPE PROG_NAME
[input ARG_STRING]
revision [true|false]
version [true|false]
promote [true|false]
icon FILENAME
checkouthistory [true|-
false];
Signatures are usually
added to the policy
afterwards:
modify policy "policy
name"
state "state name"
add signature "signature
name"
approve "user_name"
reject "user_name";
```